IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Boecker et al.**	§	
	§	Group Art Unit: **2181**
Serial No. **10/631,065**	§	
	§	Examiner: **Richard B. Franklin**
Filed: **July 31, 2003**	§	
	§	
For: **Device Address Locking to Facilitate Optimum Usage of the Industry Standard IIC Bus**	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on August 15, 2006.

A fee of $500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation, Armonk New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-7 and 15-20.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 8-14.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 1-7 and 15-20.
4. Claims allowed: None.
5. Claims rejected: 1-7 and 15-20.
6. Claims objected to: None.

C. CLAIMS ON APPEAL

The claims on appeal are: 1-7 and 15-20.

STATUS OF AMENDMENTS

An after-final response was filed on June 26, 2006. A supplemental after-final office action was filed on July 24, 2006. In an advisory action dated August 4, 2006 the examiner entered both amendments, but maintained the rejections under 35 U.S.C. §103(a).

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

Claim 1 is directed to a method for performing bus arbitration. The method includes receiving, by a device driver layer from at least one application included in an application layer (Specification, p. 11, l. 8 through p. 12, l. 9; and Figure 4, reference numerals 402, 410, and 414), a request to perform a device access operation on an end device on a bus, the device driver layer including at least one device driver that communicates with the end device utilizing the bus (Specification, p. 11, l. 8 through p. 12, l. 9), determining, by the device driver layer, whether the end device is locked (Specification p. 11, l. 8 through p. 12, l. 9), and responsive to the end device not being locked, locking, by the device driver layer, the end device and performing the device access operation (Specification, p. 11, l. 8 through p. 12, l. 9).

B. CLAIM 15 - INDEPENDENT

Claim 1 is directed to an apparatus for performing bus arbitration (Specification, p. 6, ll. 2-13). The apparatus includes a bus (Specification, p. 7, ll. 1-13; and Figure 2, reference numeral 206), at least one end device connected to the bus, at least one application included in an application layer (Specification, p. 11, l. 8 through p. 12, l. 9; and Figure 4, reference numerals 402, 410, and 414), and a driver layer that includes a wrapper layer, the driver layer including at least one device driver that communicates with the at least one end device utilizing the bus, wherein the wrapper layer receives a request from the at least one application to perform a device access operation on the at least one end device from within the at least one end device on the bus, determines whether the at least one end device is locked (Specification, p. 11, l. 8 through p. 12, l. 9), and, responsive to the at least one end device not being locked, locks the at least one end device and performs the device access operation (Specification, p. 11, l. 8 through p. 12, l. 9).

C. CLAIM 16 - INDEPENDENT

Claim 1 is directed to a computer program product, in a computer readable medium, for performing bus arbitration (Specification, p. 14, l. 15 through p. 15, l. 2). The computer program product includes instructions for receiving, by a device driver layer from at least one application

included in an application layer, a request to perform a device access operation on an end device on a bus (Specification, p. 11, l. 8 through p. 12, l. 9; and Figure 4, reference numerals 402, 410, and 414), the device driver layer including at least one device driver that communicates with the end device utilizing the bus (Specification, p. 11, l. 8 through p. 12, l. 9), instructions for determining, by the device driver layer, whether the end device is locked (Specification, p. 11, l. 8 through p. 12, l. 9), and instructions, responsive to the end device not being locked, for locking, by the device driver layer, the end device and performing the device access operation (Specification, p. 11, l. 8 through p. 12, l. 9).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. GROUND OF REJECTION 1 (Claims 8-14)

Whether claims 8-14 are indefinite under 35 U.S.C. §112.

B. GROUND OF REJECTION 2 (Claims 8-14)

Whether claims 8-14 are statutory under 35 U.S.C. §101.

C. GROUND OF REJECTION 3 (Claims 1-20)

Whether claims 1-20 are non-obvious under 35 U.S.C. §103(a) in view of *Freitas* et al., Method for Managing Concurrent Processes Using Dual Locking, U.S. Patent 6,401,110 (June 4, 2002) (hereinafter "*Freitas*") and *Burd*, Systems Architecture: Hardware and Software in Business Information Systems, 2nd ed., 1998, pp. 73-78 (hereinafter "*Burd*").

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 8-14)

The examiner rejected claims 8-14 as indefinite under 35 U.S.C. §112. Applicants canceled these claims in the supplemental response to final office action filed on July 24, 2006. Because the examiner entered that amendment in the advisory action of August 4, 2006, the rejection thereby is rendered moot.

B. GROUND OF REJECTION 2 (Claims 8-14)

The examiner rejected claims 8-14 as non-statutory under 35 U.S.C. §101. Applicants canceled these claims in the supplemental response to final office action filed on July 24, 2006. Because the examiner entered that amendment in the advisory action of August 4, 2006, the rejection thereby is rendered moot.

C. GROUND OF REJECTION 3 (Claims 1-20)

The examiner rejected claims 1-20 as non-obvious under 35 U.S.C. § 103(a) in view of *Freitas* and *Burd*. With respect to claims 8-14, the rejection is rendered moot as these claims have been canceled.

Applicants point out that the photocopy of *Burd* provided with the final office action of May 24, 2006 does not show a date. Thus, Applicants are unable to verify the publication date of *Burd* and, accordingly, are unable to verify that *Burd* can be used as a reference against the claims. The examiner indicates in the notice of references cited that the publication date of *Burd* is 1998. Applicants respectfully request that the Board verify that the publication date of *Burd* is 1998. Applicants further request the examiner to provide Applicants with a photocopy from *Burd* showing *Burd's* publication date.

Applicants assume for purposes of this appeal that *Burd* can be used as a reference. With respect to the remaining claims the examiner states that:

> As per claims 1, 8, and 16, *Freitas* teaches a method for performing bus arbitration, the method comprising receiving, by an adapter (*Freitas*; Figure 1B Items 154 and 156) from a host (*Freitas*; Figure 1B Item 152), a request to perform a device access operation (*Freitas*; Col 7 lines 37-28) on an end device (*Freitas*; Figure 1B Items 160, 162, and 1640 on a bus (*Freitas*;

Figure 1B Item 158); determining, by the adaptor, whether the end device is locked (*Freitas*; Figure 4 Item 407, Col 7 Lines 38-45); and responsive to the end device not being locked, locking (*Freitas*; Figure 4 Items 408 and 410, Col 7 Lines 38-45, Col 8 Lines 1-12), by the adaptor, the end device and performing the device access operation (*Freitas*; Figure 4 Item 412).

Freitas does not teach wherein the adaptor is a device driver layer, the host is an application included in an application layer, and the device driver layer includes a device driver that is in communication with the end device utilizing the bus.

However, Burt teaches a computer architecture and operating system where an application program (host) must use a device driver in a kernel layer (device driver layer) to access a hardware device (*Burd*; Figure 3-7; Pages 73-74 "Operating System Model and Functions"). The device driver is contained in a kernel layer and is used to interface with the hardware device (*Burd*; Pages 76-77 "Kernel").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of *Freitas* to include the application and device driver layers because using the layers allows for a degree of modularity and hardware independence (*Burd*; Pages 76-77 "Kernel").

Final office action of May 24, 2006, pp. 5-6.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). A *prima facie* case of obviousness is established when the teachings of the prior art itself suggest the claimed subject matter to a person of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 783, 26 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1993). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). For an invention to be *prima facie* obvious, the prior art must teach or suggest all claim limitations. *In re Royka,* 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In the case at hand, the cited references do not teach or suggest all of the limitations of the claims, arranged as they are in the claims.

Claim 1 is a representative claim of claims 1-7 and 15-20. Claim 1 is as follows:

> 1. A method for performing bus arbitration, the method comprising: receiving, by a device driver layer from at least one application included in an application layer, a request to perform a device access operation on an

end device on a bus, the device driver layer including at least one device driver that communicates with the end device utilizing the bus;

determining, by the device driver layer, whether the end device is locked; and

responsive to the end device not being locked, locking, by the device driver layer, the end device and performing the device access operation.

C.1. The Proposed Combination Does Not Teach All of the Features of Claim 1

The examiner failed to state a *prima facie* obviousness rejection against claim 1 because the proposed combination of *Freitas* and *Burd*, when considered as a whole, does not teach or suggest all of the features of claim 1. Specifically, the proposed combination does not teach or suggest the claimed feature of determining, by the device layer, whether the *end device is locked* and locking, by the device driver layer, *the end device*.

The examiner asserts otherwise, citing the various portions of *Freitas* as teaching these claimed features. Applicants address each citation in turn. First, the examiner refers to Figure 4. Figure 4 of *Freitas* is as follows:

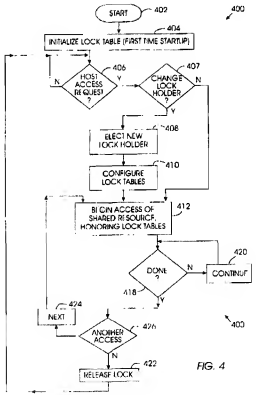

FIG. 4

The examiner specifically refers to reference numerals 407, 408, and 410 in Figure 4 as teaching the claimed features of determining and locking, as recited in claim 1. Reference numeral

407 refers to a change in lock holder decision block. Reference numeral 408 refers to an elect new lock holder block. Reference numeral 410 refers to a configure lock tables block.

In each case, *Freitas* refers to lock tables and establishment of a "lock holder." As shown further below, the lock holders and lock tables are stored in adaptors, such as adaptors 154 and 156 in Figure 1. The lock holders and lock tables cause the processors in *Freitas* to "honor" a lock state. See, for example, the following portion of *Freitas*;

> Multiple competing processors cooperatively manage access to a shared resource. Each processor separately stores a lock table, listing shared resource subparts, such as memory addresses of a data storage device, for example. The lock tables are stored in nonvolatile storage. In each lock table, each subpart is associated with a "state," such as; LOCAL or REMOTE. In response to access requests from the hosts, the processors exchange various messages to cooperatively elect a single processor to have exclusive access to the subparts involved in the access requests. After one processor is elected, the lock-holding processor configures its lock table to show the identified subpart in the LOCAL state, and all non-lock-holding processors configure their lock tables to show the identified subpart in the REMOTE state. Thus, rather than replicating one lock table for all processors, the processors separately maintain lock tables that are coordinated with each other. Importantly, each processor honors its lock table by refraining from accessing a subpart of the shared resource unless the processor's lock table indicates a LOCAL state for that subpart. In one embodiment, optimized for the two processor environment, the messages exchanged by the processors include lock request, lock release, and lock grant messages.

Freitas, Abstract.

Thus, *Freitas* creates a mechanism by which processors *avoid accessing* a shared resource. In contrast, claim 1 requires "determining, by the device driver layer, whether *the end device* is locked" and "responsive to the end device not being locked, *locking*, by the device driver layer, *the end device* and performing the device access operation" (emphasis added). Thus, claim 1 requires that the *device itself* be locked and unavailable. This fact is supported by the specification, which provides in part that:

> Abstract wrapper layer **412** determines whether the device is available. If the device is available, the abstract wrapper layer *places a lock on the device address*, restricting its use by any other application. In the meantime, other applications that request access to any other device on the same bus

> can request a lock on that device. The device driver layer keeps a list **413** of occupied end device addresses and is responsible for *locking the resources* when transfers are ongoing and for releasing resources when transfers are complete.

Specification, p. 12, ll. 1-10 (emphasis added).

Maintaining a table in order to prevent a processor from voluntarily accessing an end device is different than locking the end device itself. For this reason, Figure 4 does not teach the claimed features of "determining, by the device driver layer, whether *the end device* is locked" and "responsive to the end device not being locked, *locking*, by the device driver layer, *the end device* and performing the device access operation" (emphasis added). Because *Freitas* is devoid of disclosure in this regard, and because the claimed invention is counter-intuitive to the teachings of *Freitas*, *Freitas* also does not suggest these claimed features.

Now addressing the next portion of *Freitas*, the examiner also refers to the following portion of *Freitas*:

> After step 404, the adapters 154, 156 await shared resource access requests from the hosts 152 (step 406). *In response to such a request, which involves one or more identified subparts, the adapters 154, 156 cooperatively determine whether there should be any change in the lock holder (step 407). For instance, if no adapter holds the lock on the requested subparts, a change in the lock holder is needed to give the lock to the adapter that received the access request.* In another example, the processor already owning the lock may maintain a queue of lock requesting tasks, including its own and lock requests from the other processor. Whenever the other processor's lock requests arrives at the head of the queue, there is a need for a change in the lock holder, i.e., to provide the lock to the lock requesting processor.

Freitas, col. 7, ll. 37-51 (emphasis to show portions cited by the examiner).

Again, the cited portion of *Freitas* teaches that the adaptors prevent the processors from accessing the protected resources. However, the resources themselves are not locked, as recited in claim 1. *Freitas* does refer to the lock holder (step 407), which is described in the examiner's next citation as follows:

> In contrast to the foregoing sequence, whenever step 407 determines that the lock holder must be changed, the adapters 154, 156 effectively change the lock holder in step 408. In the case of two adapters 154, 156 (as illustrated), this involves electing the non-lock-holding adapter to be the new lock holder. In the case of three or more processors, the processors may

> cooperatively elect a new lock holder using a suitably fair arbitration scheme, such as enqueing adapters' lock requests and processing the requests sequentially, etc. In response to step 408, the adapters configure their respective lock tables to reflect the newly elected lock holder, as shown by step 410.

Freitas, col. 8, ll. 1-12.

Freitas specifically states that adaptors hold the locks and that the processors cooperatively *elect* a new lock holder. Nowhere does *Freitas* teach or suggest that the end device itself is locked, as claimed. Moreover, *Freitas* is devoid of disclosure in this regard. Therefore, *Freitas* does not teach or suggest this claimed feature.

Additionally, *Burd* does not teach or suggest the claimed features of "determining, by the device driver layer, whether *the end device* is locked" and "responsive to the end device not being locked, *locking*, by the device driver layer, *the end device* and performing the device access operation" (emphasis added). The examiner does not assert otherwise. Instead, *Burd* describes the basic architecture of operating systems. *Burd* teaches that the kernel is the portion of the operating system that interacts with hardware. Thus, *Burd* teaches the same thing that *Freitas* teaches – that the processors or adaptors are prevented from attempting access to the end devices. *Burd* does not teach or suggest otherwise.

In contrast, the above features of claim 1 require locking the end devices themselves. *Burd* does not teach or suggest these claimed features because *Burd* is devoid of disclosure in this regard. Therefore, neither *Freitas* nor *Burd* teach the claimed features of "determining, by the device driver layer, whether *the end device* is locked" and "responsive to the end device not being locked, *locking*, by the device driver layer, *the end device* and performing the device access operation" (emphasis added). Accordingly, the proposed combination, when considered as a whole, does not teach or suggest all of the features of claim 1. Therefore, under the standards of *In re Royka*, the examiner has failed to state a *prima facie* obviousness rejection against claim 1.

C.2. The Examiner Failed To State a Proper Teaching, Suggestion, or Motivation to Combine the References

A proper *prima facie* case of obviousness cannot be established by combining the teachings of the prior art absent some teaching, incentive, or suggestion supporting the combination. *In re*

Napier, 55 F.3d 610, 613, 34 U.S.P.Q.2d 1782, 1784 (Fed. Cir. 1995); *In re Bond*, 910 F.2d 831, 834, 15 U.S.P.Q.2d 1566, 1568 (Fed. Cir. 1990). In the case at hand the examiner has failed to establish a proper teaching, incentive, or suggestion supporting the combination and no such teaching, incentive, or suggestion exists.

The examiner states that:

> Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of *Freitas* to include the application and device driver layers because using the layers allows for a degree of modularity and hardware independence (*Burd*; Pages 76-77 "Kernel").

Final office action of May 24, 2006, p. 5-6.

This statement ignores features of claim 1; specifically, "determining, by the device driver layer, whether *the end device* is locked" and "responsive to the end device not being locked, *locking*, by the device driver layer, *the end device* and performing the device access operation" (emphasis added). As shown above, the proposed combination of references, when considered as a whole, do not teach or suggest these claimed features. The examiner's statement fails to account for this fact. Therefore, the examiner has failed to state a *prima facie* obviousness rejection.

C.3. The References Teach Away from Claim 1

The examiner failed to state a *prima facie* obviousness rejection because no teaching, suggestion, or motivation exists to combine the references to achieve the invention of claim 1 under the standards of *In re Napier*. No teaching, suggestion, or motivation exists to combine the references *Freitas* and *Burd* specifically teach away from the invention of claim 1.

As shown above, *Freitas* and *Burd* teach using adaptors, software, or some other mechanism to prevent processors from accessing end devices. In contrast, claim 1 requires that the end devices themselves be locked in the manner described in claim 1. *Freitas* and *Burd* therefore specifically teach a method of protecting end devices from conflict that is contrary to the claimed method. For this reason, one of ordinary skill would be motivated to avoid combining the references to achieve the invention of claim 1. Accordingly, no teaching, suggestion, or motivation exists to combine the references to achieve the invention of claim 1. Therefore, the examiner has failed to state a *prima facie* obviousness rejection against claim 1.

C.4. The Proposed Combination Changes the Principle of Operation of the Primary Reference

Further regarding claim 1, the examiner has failed to state a *prima facie* obviousness rejection because the proposed combination changes the principle of operation of the primary reference. In combining references to show the claimed feature, the proposed modification cannot change the principle of operation of a reference. See *In re Ratti*, 270 F.2d 810, 123 (CCPA 1959) and MPEP 2143.01. If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. Id.

In the case at hand, the proposed combination of the references changes the principle of operation of the invention of claim 1. *Freitas* uses adaptors to prevent processor access to end devices. Specifically, *Freitas* provides that:

> In step 404, the adapters 154, 156 initialize their respective lock tables 172, 174 if needed. Namely, initialization is necessary of this is a first time startup, where no previous lock tables exist. In one embodiment, this may involve allocating storage for the lock tables 172, 174, preparing pointers, and performing other storage tasks to ready the tables for use. If desired, step 404 may additionally prepare blank entries in the lock table, where each entry corresponds to the minimum size of separately accessible shared resource subpart, such as a single address, partition, etc.
>
> After step 404, the adapters 154, 156 await shared resource access requests from the hosts 152 (step 406). In response to such a request, which involves one or more identified subparts, the adapters 154, 156 cooperatively determine whether there should be any change in the lock holder (step 407). For instance, if no adapter holds the lock on the requested subparts, a change in the lock holder is needed to give the lock to the adapter that received the access request. In another example, the processor already owning the lock may maintain a queue of lock requesting tasks, including its own and lock requests from the other processor. Whenever the other processor's lock requests arrives at the head of the queue, there is a need for a change in the lock holder, i.e., to provide the lock to the lock requesting processor.

Freitas, col. 7, ll. 27-51.

Thus, *Freitas* teaches that adaptors provide the lock holds on end devices. The entire principle of operation of *Freitas'* system is to provide a means for locking the end devices without using software, as shown by the following portion of *Freitas*:

> One recurring challenge to systems with multiple processors involves the sharing of resources by the multiple processors. As one example, digital data storage such as magnetic "hard" disk drive storage is often shared by multiple storage "adapters." Sharing such a resource is challenging because of the difficulties in arbitrating access to the resource. At any given time, which processor should be permitted access to the shared resource? Should other processors be given limited concurrent access? This is further complicated by the need to plan for possible failure of a processor or communications between the processors.
>
> One popular approach to sharing computer resources is called "mutual exclusion," which is often applied at the device level. With this approach, processors access the resource one-at-a-time. While one processor is accessing the resource, all other processors are excluded from that device. Although this approach is attractive in its simplicity, shared computer resources often possess significantly more input/output ("I/O") capability than the processors that manage them. In this case, the full throughput of the shared resource is wasted when it is being used by one processor to the exclusion of the other processors.
>
> In the case of storage resources, the system takes longer to store and retrieve data when the processors are confined by one-at-a-time access rules. This is undesirable, since slower data storage and retrieval are frustrating to most computer users. Furthermore, slow data access may be intolerable in certain data-critical applications, such as automated teller networks, airline reservation systems, stock brokerage, etc. Furthermore, the use of mutual exclusion is complicated by the possibility that a processor with exclusive access to the shared resource experiences a failure, causing a severe problem for the excluded processors.
>
> To orchestrate mutual exclusion, competing processors must exchange messages of some type. A different set of problems is thus presented by the possibility that messages are lost while a device is reserved to one processor, causing a situation known as "livelock." A further difficulty inherent to mutual exclusion schemes is the need to fairly allocate access to the shared resource among competing processors, the consequences of misallocation potentially including "starvation" of the losing processor.

Freitas, col. 1, ll. 22-64.

In contrast, *Burd* provides that basic operating system software controls access to end devices via the software. The examiner's proposed combination changes the principle of operation of *Freitas'* system because, in the examiner's proposed combination, the *Freitas* adaptors would be replaced by an operating system. *Freitas* seeks to avoid this implementation. Thus, the examiner's

proposed combination changes the principle of how *Freitas* controls processor access to resources.

As shown above, *In re Ratti* provides that the teachings of the references are not sufficient to render the claims *prima facie* obvious if the combination changes the principle of operation of the primary reference. Therefore, the examiner has failed to state a *prima facie* obviousness rejection against claim 1.

C.5. No Teaching, Suggestion, or Motivation Exists to Combine the References in the Proposed Manner

Additionally, the examiner failed to state a *prima facie* obviousness rejection because no teaching, suggestion, or motivation exists to combine the references in the proposed manner. As shown above, *Burd* provides for a basic operating system architecture in which the operating system controls how the processor or processors access end devices. However, also as shown above, *Freitas* specifically desires to avoid that implementation. Instead, *Freitas* teaches the use of adaptors to control processor access to end devices.

Thus, one of ordinary skill would, upon reading *Freitas*, reject any notion of combining *Burd* with *Freitas* to implement control of processor access to end devices. Accordingly, no teaching, suggestion, or motivation exists to combine the references. Therefore, the examiner failed to state a *prima facie* obviousness rejection against claim 5.

C.6. Remaining Claims

The remaining claims contain features similar to those presented in claim 1 or depend from claim 1. Therefore, the examiner failed to state a *prima facie* obviousness rejection against these claims for the same reasons presented above vis-à-vis the response to the rejection of claim 1.

D. CONCLUSION

As shown above, the examiner has failed to state a *prima facie* obviousness rejection against any of the claims. Therefore, Applicants request that the Board of Patent Appeals and Interferences reverse the rejections. Additionally, Applicants request that the Board direct the examiner to allow the claims.

/Theodore D. Fay III/
Theodore D. Fay III
Reg. No. 48,504
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A method for performing bus arbitration, the method comprising:

receiving, by a device driver layer from at least one application included in an application layer, a request to perform a device access operation on an end device on a bus, the device driver layer including at least one device driver that communicates with the end device utilizing the bus;

determining, by the device driver layer, whether the end device is locked; and

responsive to the end device not being locked, locking, by the device driver layer, the end device and performing the device access operation.

2. The method of claim 1, wherein the device access operation is one of a read operation and a write operation.

3. The method of claim 1, further comprising:

responsive to the end device being locked, denying the device access operation.

4. The method of claim 1, wherein the step of determining whether the end device is locked includes determining whether an address of the end device is found in a list of occupied end devices.

5. The method of claim 1, wherein the step of locking the end device includes placing a device address of the end device in a list of occupied end devices.

6. The method of claim 5, further comprising:

responsive to the device access operation completing, unlocking the end device.

7. The method of claim 6, wherein the step of unlocking the end device includes removing the device address from the list of occupied end devices.

15. An apparatus for performing bus arbitration, the apparatus comprising:

a bus;

at least one end device connected to the bus;

at least one application included in an application layer; and

a driver layer that includes a wrapper layer, the driver layer including at least one device driver that communicates with the at least one end device utilizing the bus,

wherein the wrapper layer receives a request from the at least one application to perform a device access operation on the at least one end device from within the at least one end device on the bus, determines whether the at least one end device is locked, and, responsive to the at least one end device not being locked, locks the at least one end device and performs the device access operation.

16. A computer program product and a computer readable medium in which the computer program product is stored, for performing bus arbitration, the computer program product comprising:

instructions for receiving, by a device driver layer from at least one application included in an application layer, a request to perform a device access operation on an end device on a bus, the device driver layer including at least one device driver that communicates with the end device utilizing the bus;

instructions for determining, by the device driver layer, whether the end device is locked; and

instructions, responsive to the end device not being locked, for locking, by the device driver layer, the end device and performing the device access operation.

17. The computer program product of claim 16, wherein the instructions for determining whether the end device is locked include instructions for determining whether an address of the end device is found in a list of occupied end devices.

18. The computer program product of claim 16, wherein the instructions for locking the end device include instructions for placing a device address of the end device in a list of occupied end devices.

19. The computer program product of claim 18, further comprising:

instructions, responsive to the device access operation completing, for unlocking the end device.

20. The computer program product of claim 19, wherein the instructions for unlocking the end device include instructions for removing the device address from the list of occupied end devices.

EVIDENCE APPENDIX

No additional evidence is presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.